

PDF del Código de Planning con MaríaDB

Demostración de uso de MariaDB

```
// Importar módulos necesarios
const express = require('express');
const mysql = require('mysql');
const app = express();
const bodyParser = require('body-parser');

// Configurar middleware para manejar datos JSON y URL-
encoded
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// Configura la conexión a la base de datos MySQL
const db = mysql.createConnection({
  host: '127.0.0.1',
  user: 'root',
  password: '1234',
  database: 'planning',
  connectionLimit: 3,
});

// Se conecta a la base de datos y realiza modificaciones
necesarias
db.connect((err) => {
  if (err) {
    console.log('Error al conectar a la base de datos:', err);
  } else {
    console.log('Conexión exitosa a la base de datos');

    // Modificar la tabla para permitir valores nulos en created_at
    db.query('ALTER TABLE todo MODIFY created_at
TIMESTAMP NULL', (alterErr) => {
```

```

        if (alterErr) {
            console.error('Error al modificar la tabla:', alterErr);
        } else {
            console.log('Tabla modificada correctamente');
        }
    });

    // Modificar la tabla para permitir valores nulos en
    updated_at
    db.query('ALTER TABLE todo MODIFY updated_at
    TIMESTAMP NULL', (alterErr) => {
        if (alterErr) {
            console.error('Error al modificar la tabla:', alterErr);
        } else {
            console.log('Tabla modificada correctamente');
        }
    });
}
});

// Endpoint: Obtener todos los elementos
app.get('/todo', (req, res) => {
    db.query('SELECT * FROM todo', (err, result) => {
        if (err) {
            console.error('Error al obtener elementos:', err);
            res.send({ success: false, error: err.message });
        } else {
            res.send({ success: true, data: result });
        }
    });
});

// Endpoint: Obtener un elemento por ID
app.get('/todo/:id', (req, res) => {
    const todold = req.params.id;

```

```

// Verificar si el ID proporcionado es un número válido
if (isNaN(todoId)) {
    return res.status(400).send({ success: false, error: 'ID debe
ser un número válido' });
}

// Realizar la consulta para obtener el elemento por ID
db.query('SELECT * FROM todo WHERE id = ?', [todoId], (err,
result) => {
    if (err) {
        console.error('Error al obtener el elemento por ID:', err);
        res.send({ success: false, error: err.message });
    } else {
        // Verificar si se encontraron resultados
        if (result.length === 0) {
            res.status(404).send({ success: false, error: 'Elemento
no encontrado' });
        } else {
            res.send({ success: true, data: result[0] });
        }
    }
});
});

```

```

// Endpoint: Añadir un nuevo elemento
app.post('/todo', (req, res) => {
    const { name, description, status } = req.body;
    const created_at = new Date(); // Lógica para obtener la fecha
y hora actual
    const update_at = new Date(); // Lógica para obtener la fecha
y hora actual
    db.query(
        'INSERT INTO todo (name, description, created_at,
update_at, status) VALUES (?, ?, ?, ?, ?)',

```

```

    [name, description, created_at, update_at, status],
    (err, result) => {
      if (err) {
        console.error('Error al añadir un nuevo elemento:', err);
        res.status(500).send({ success: false, error: 'Error
interno del servidor' });
      } else {
        res.status(201).send({ success: true, data: result });
      }
    }
  );
});

```

// Endpoint: Eliminar un elemento por ID

```

app.delete('/todo/:id', (req, res) => {
  const todoid = req.params.id;

  db.query('DELETE FROM todo WHERE id = ?', [todoid], (err,
result) => {
    if (err) {
      console.error('Error al eliminar elemento:', err);
      res.send({ success: false, error: err.message });
    } else {
      res.send({ success: true, message: 'Elemento eliminado
correctamente' });
    }
  });
});

```

// Maneja la señal de interrupción para cerrar la conexión a la base de datos al cerrar la aplicación

```

process.on('SIGINT', () => {
  db.end((err) => {
    if (err) {

```

```
        console.error('Error al cerrar la conexión a la base de
datos:', err);
    } else {
        console.log('Conexión a la base de datos cerrada');
        process.exit();
    }
});
});
```